Pittsburgh, PA 15213-3890

Towards a Reference Model of Integration and Interoperability

Patrick Kirwan

September 22, 2005

Sponsored by the U.S. Department of Defense

Outline

Background of “the interoperability problem”

Current models of interoperability

Proposed characteristics of a unified model

Conclusion

Current State of Software Engineering

New systems usually a heterogeneous collection of custom and commercial products

- Integration provided by some third-party technology

New systems seldom expected to function independently

- Expected to cooperate with existing systems
- Ability to achieve “cooperation” is generally termed "interoperability"

Elements of these cooperating systems undergo frequent changes (e.g., upgrades of commercial products)

Thus: boundaries within and between systems begin to blur

- Distinction between a "system of systems" and a single, complex, distributed system disappears

Current State - 2

Interoperability can occur only when some degree of compatibility exists among all elements that must cooperate for some purpose

Interoperability is based on the existence of (and cannot occur lacking) a single, common conceptual view

- Conceptual view can be embodied in an architecture or design
- Conceptual view can be implemented through a common protocol
- Single conceptual view determines whether a system (or system-of-systems) can made to cooperate as intended

The Problem Space

Incomplete understanding of scope and nature of the engineering to be accomplished

- Cannot discern incompatible solutions or intractable problems

Ongoing inertia toward separate programs, managed and executed independently

- Cannot, in such a climate, ensure that independent programs act in service of a common goal (i.e., the interoperable end goal)

Few technologies currently exist that permit quantification of any aspect of interoperability

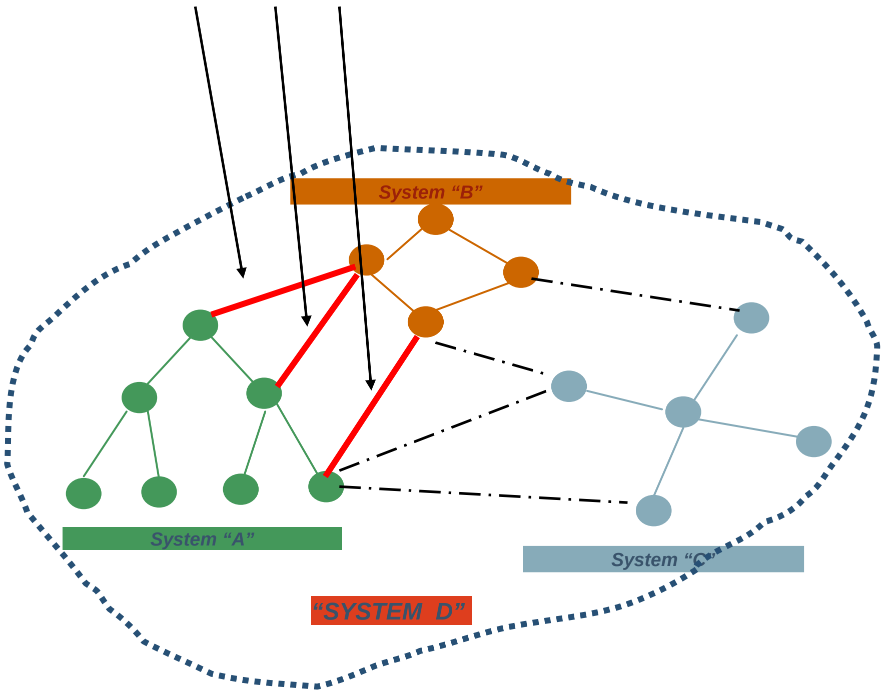
An Instance of the Problem

We know quite a lot about constructing systems from components (over which we may have little or no control).

We know something about composing *systems of systems* from individual systems (over which we may have little or no control).

We know very little about constructing an *interoperable network of systems*…the key distinction being that the network is unbounded (or very loosely bounded) and has no single controlling authority.

Unplanned, unexpected, emergent behavior here…



ISIS - Purpose and Vision

Integration of Software-Intensive Systems

Identify and mature software engineering methods and techniques for broad-based and sustained integration and interoperability across components, systems, and systems-of-systems

Identify best practice for interoperability and integration to DoD and other organizations as their systems migrate to an increasingly net-centric world.

Outline

Background of “the interoperability problem”

Current models of interoperability

Proposed characteristics of a unified model

Conclusion

NATO C3 SAF Model

NATO C3 System Architecture Framework (NC3SAF)

- Mandated for NC3 systems architectures.
- Includes three main types of guidance for architecture development
  - Guidelines that include guiding principles for building architectures
  - Process to build and integrate architectures
  - Templates with detailed descriptions.

Based on the DoD C4ISR Architectural Framework

- Different from its U.S. counterpart in that it is inclusive of specific NATO directives, precepts and tenets.

Includes an extensive discussion of interoperability

NATO - 2

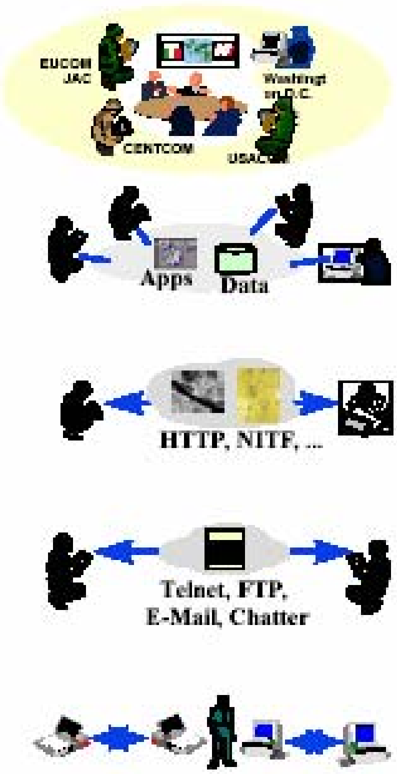
Levels of interoperability:

- **No Data Exchange**
  - No physical connection exists
- **Unstructured Data Exchange**
  - Exchange of human-interpretable, unstructured data (free text)
- **Structured Data Exchange**
  - Exchange of human-interpretable structured data intended for manual and/or automated handling, but requires manual compilation, receipt, and/or message dispatch
- **Seamless Sharing of Data**
  - Automated data sharing within systems based on a common exchange model
- **Seamless Sharing of Information**
  - Universal interpretation of information through cooperative data processing

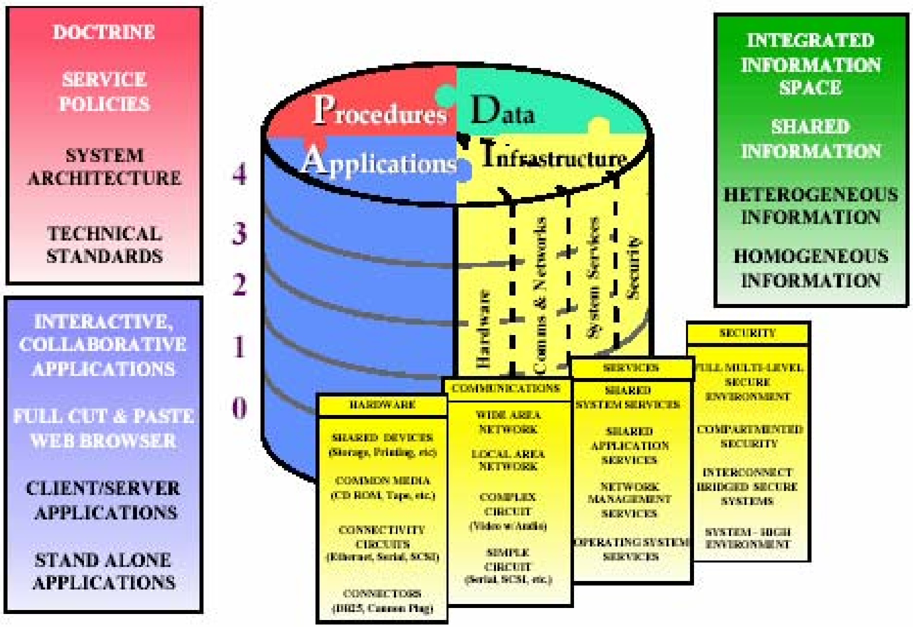
LISI Model: Levels of Interoperability

Information Exchange	*Level*
Distributed global info. and apps. **Simultaneous interactions w/ complex data** **Advanced collaboration** e.g., Interactive COP update Event-triggered global database update	**4 -- Enterprise** Interactive manipulation Shared data & applications
Shared databases **Sophisticated collaboration** e.g., Common Operational Picture	**3 -- Domain** Shared data "Separate" applications
Heterogeneous product exchange **Group Collaboration** e.g., Exchange of annotated imagery, maps w/ overlays	**2 -- Functional** Minimal common functions Separate data & applications
Homogeneous product exchange e.g., FM voice, tactical data links, text file transfers, messages, e-mail	**1 -- Connected** Electronic connection Separate data & applications
Manual Gateway e.g., diskette, tape, hard copy exchange	**0 -- Isolated** Non-connected

Computing Environment



LISI Model: “PAID” Attributes

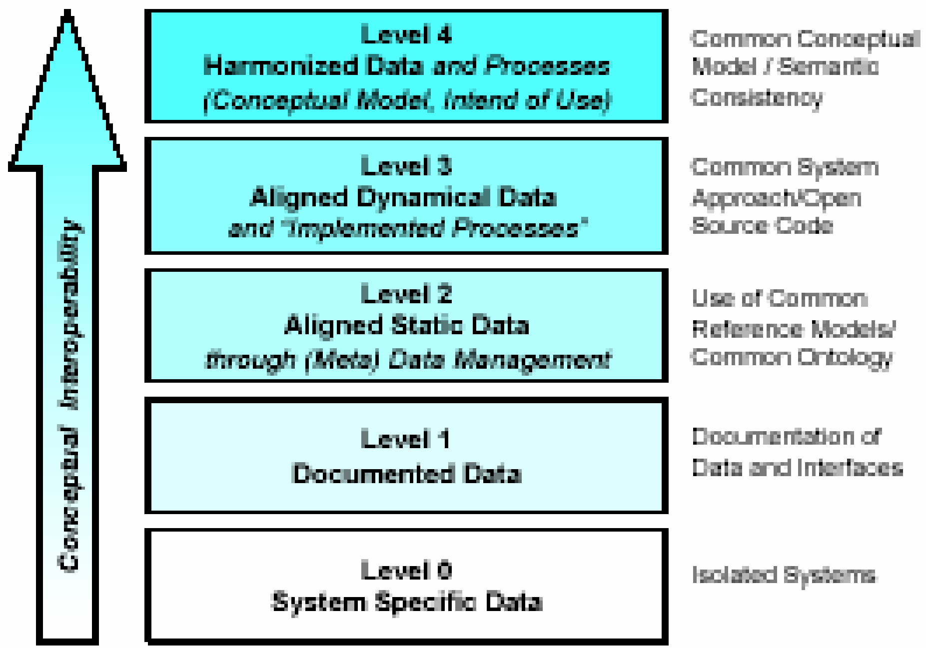


A “level” is enabled by a specific profile of P, A, I, & D attributes

LCIM model

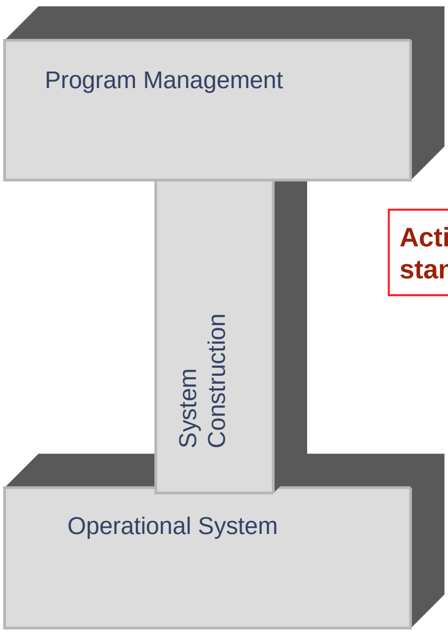
Incorporates notion of Conceptual interoperability

- Explicit focus on semantic issues
- Maintains concept of increasing maturity, levels, etc.



SOSI model

Focus is on different domains of interoperability
- **Programmatic, Constructive, Operational**
- **Different kinds of activities and relationships in each domain**



Activities performed to manage the acquisition of a system. Focus is on contracts, incentives, and practices such as risk management.

Activities performed to create and sustain a system. Focus is on architecture, standards, and commercial off-the-shelf (COTS) products.

Activities performed to operate a system. Focus is on interactions with other systems and with users.

Outline

Background of “the interoperability problem”

Origins in integration research

Current models of interoperability

Proposed characteristics of a unified model of interoperability

Conclusion

Some General Precepts

"Interoperability" is a multi-dimensional aspect of system engineering

- Scope is far greater than simply interoperability of data
- Encompasses interoperablity at the programmatic (and other) levels
- A model must includes degrees of coupling, heterogeneity, synchronicity, . . .

We can never anticipate fully the boundaries that a given system will be expected to operate within

Interoperability must be quantifiable to be achievable

Interoperability must be sustainable and sustained

Proposed characteristics

Based on observations about desired types and levels of interoperability

Must be verified and validated through scenarios drawn from real programs

Characteristics chosen are not necessary discrete

List needs refinement through further research

Proposed characteristics - 2

Six principal characteristics:

- Coupling
- Heterogeneity
- Synchronicity
- Boundedness
- Ownership
- Usage patterns

May be more characteristics

- These may be at a lower (or higher) level of importance

Coupling

Should permit modeling the aggregate degree of coupling in an interoperating system

- Coupling among its elements (i.e., systems)
- Elements may themselves be collections of systems
- Continues recursively until some base level of complexity of internal coupling within an individual system

Aggregate degree of coupling has implications for techniques, strategies, difficulty, etc. to create, use, or sustain the entire system of systems.

Heterogeneity

Should permit modeling both syntactic and semantic complexity

- Each pair-wise set of systems will exhibit both kinds
- As the number of systems grows beyond a pair, this complexity grows combinatorially

The degree of heterogeneity (and at both syntactic and semantic levels) may suggest the degree of difficulty in achieving and sustaining interoperability between the pair.

Synchronicity

Should permit modeling the rates at which elements (i.e., individual systems) undergo change

- Change includes update, modernization, repair, and so forth
- Like other properties, this is recursive down to the level of individual components

The degree to which individual systems' rates of change are synchronous will affect the degree to which the aggregate interoperability is sustainable (and perhaps achievable at all).

Boundedness

Should permit modeling the degree and nature of external and internal system boundaries

- Some interoperable systems occurs when the aggregate collection of systems is initially known
- Other interoperable systems, actual extent of the system-of-systems is known to be unknown.

Methods, techniques, and technologies used to bring about the aggregate interoperation will likely be different

- Ongoing maintenance of the overall system will also differ

Ownership

Should permit modeling the different qualities of authority over systems and elements of systems

- Some complex systems of systems are methodically planned (e.g., U.S. DoD's Future Combat System)
  - Possible (or should be) to identify some controlling agency of the overall system(s)
- Some interoperability occurs opportunistically, when two (or more) diverse systems are linked in unplanned but useful ways
  - Usually impossible to identify any agency with responsibility for the overall aggregate system.

Will generally be very different processes, techniques, and methods used to bring about the interoperability between the constituent systems.

Usage patterns

Should permit modeling the conformity between intended and actual usage patterns throughout the system

- All elements of any system (i.e., components, entire systems) have an intended pattern of use

- An interoperating set of systems also has an intended pattern of use
  - This will conform to usage patterns of some elements, and conflict with usage patterns of other elements

Aggregate degree of harmony and conflict may determine the usability and robustness of the overall system

- This characteristics will be inconsistent across the system's elements

Outline

- Background of “the interoperability problem”
- Origins in integration research
- Current models of interoperability
- Proposed characteristics of a unified model
- **Conclusion**

Conclusion

Appropriate models have proven to be of considerable value in many engineering domains

We are presently in need of such models for integrating collections of software systems

Current efforts have produced several interesting and useful models

- Much more work is needed

Conclusion - 2

Trend toward ever-increasing interconnection between systems will continue

Nature and quality of these interconnections will be governed by decisions now being made

Effects of these decisions may be long-lasting

Questions?

Patrick Kirwan
Software Engineering Institute Europe
Carnegie Mellon University
pkirwan@sei.cmu.edu